

116

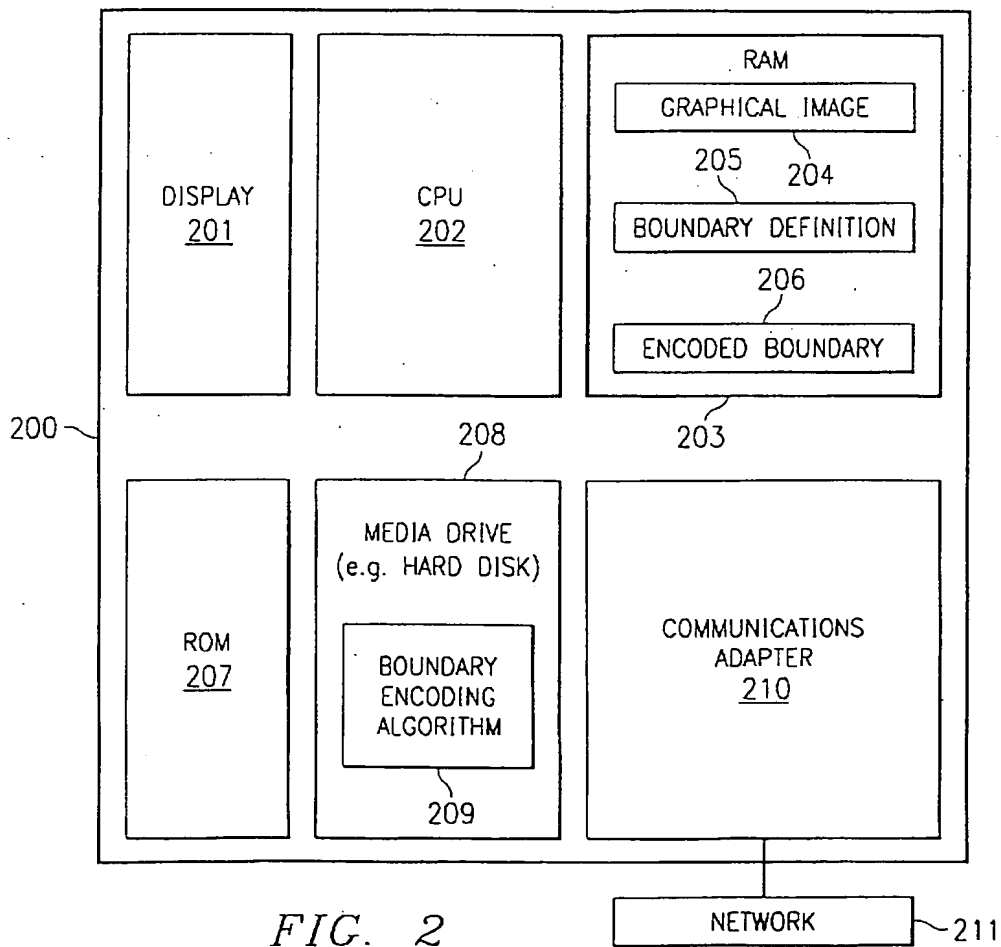
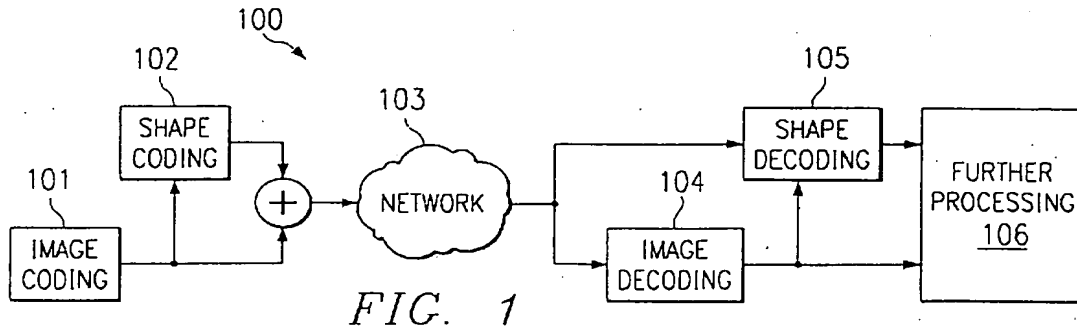


FIG. 1

2 / 6

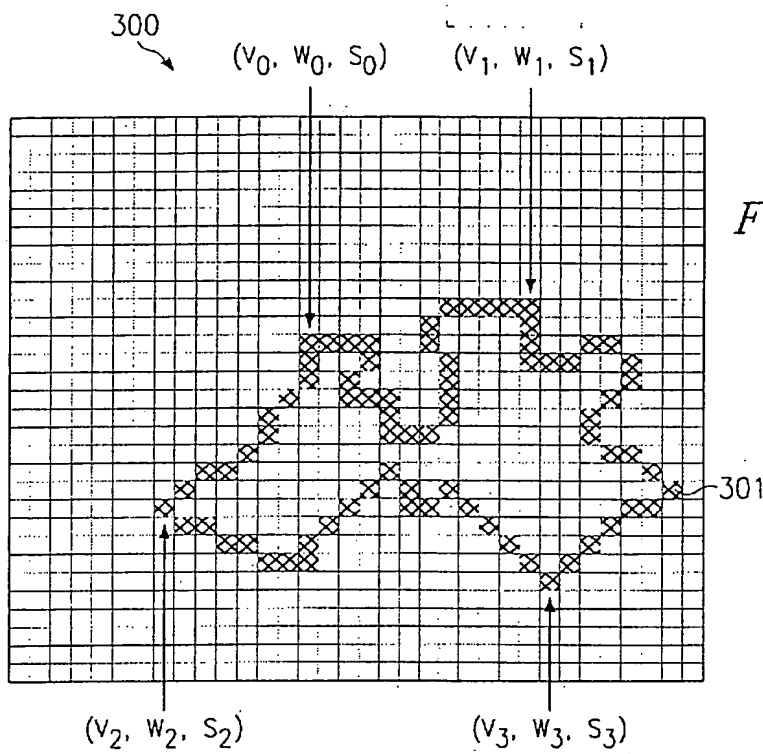


FIG. 3

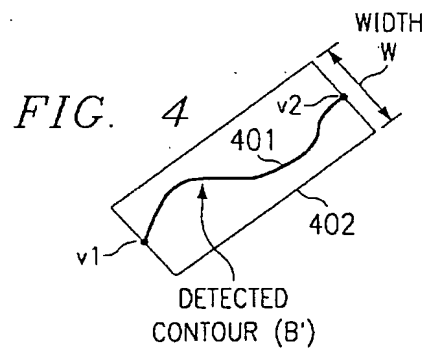


FIG. 4

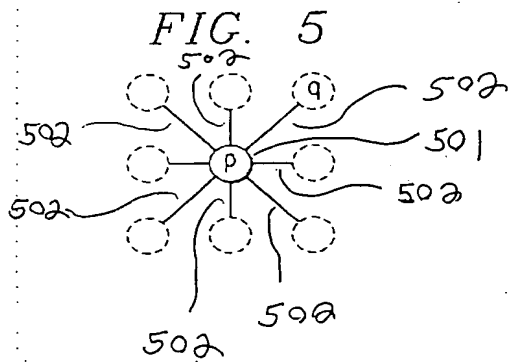
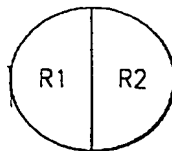


FIG. 5

FIG. 11



3 / 6

FIG. 6

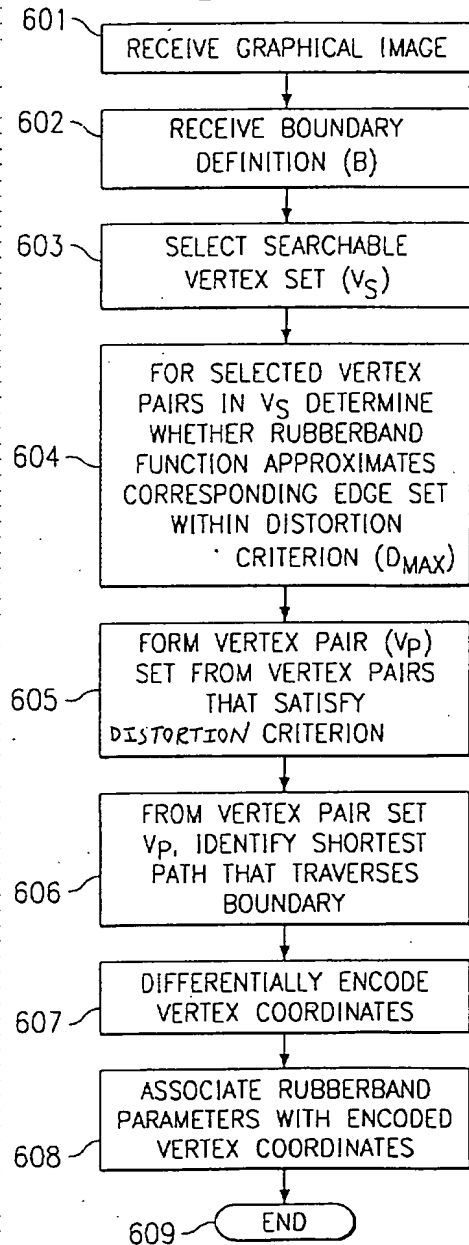


FIG. 7

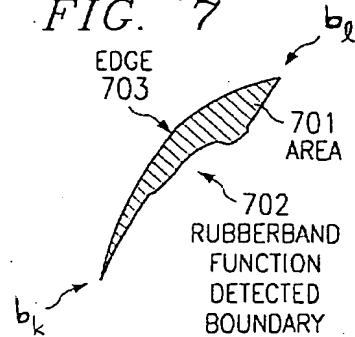
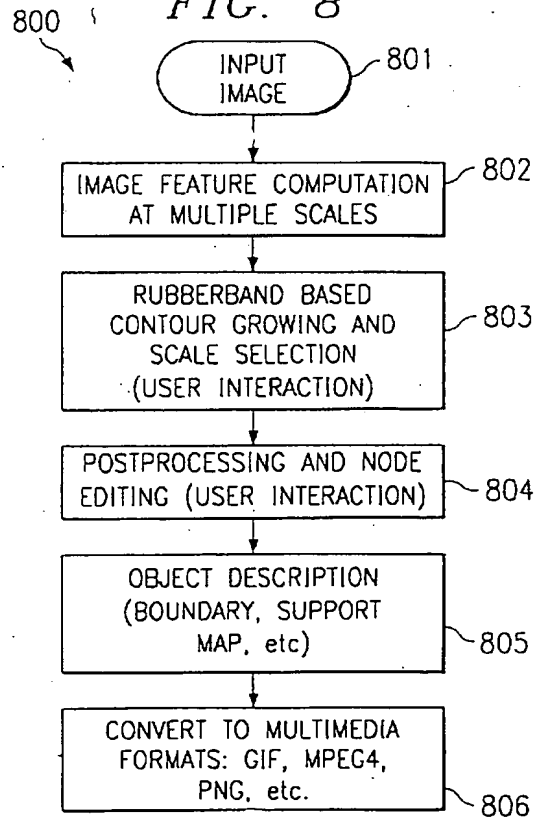


FIG. 8



4/6

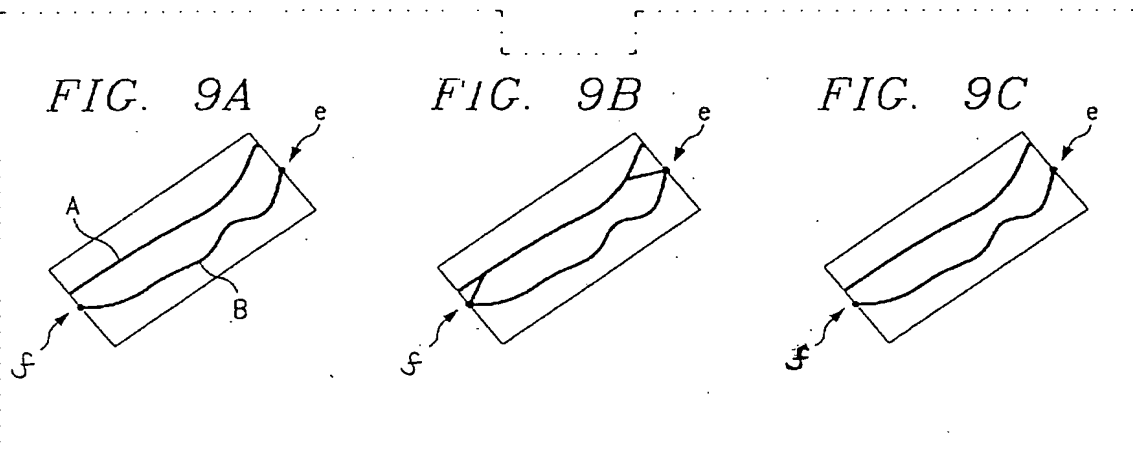


FIG. 9A

FIG. 10

Input: f (start point), e (ending point), $\text{Dist}(p,q)$ (local distance definition)

Assistant Data Structure:

$L1$ (active list 1)

$L2$ (active list 2)

$C(p)$ (cumulative distance from f to p)

Output: ptr (minimal cost path pointers)

Algorithm:

(1001) Initialize assistant data structure ($L1$, $L2$ are set empty, and C to $+\infty$).

(1002) Set initial threshold T_0 and increasing step δ_T .

(1003) $T = T_0$;

(1004) $\text{push}(L1, f, 0)$;

(1005) while($T \leq T_{\max}$ and $C(e) \neq +\infty$) {

(1006) while($\text{num}(L1) > 0$) {

(1007) $\text{pop}(L1, p)$;

(1008) $\text{flag_thresholded} = 0$;

(1009) for (each $q \in N(p)$) {

(1010) if($\text{Dist}(p, q) > T$) {

(1011) $\text{flag_threshold} = 1$; continue;

(1012) }

(1013) $d' = C(p) + \text{Dist}(p, q)$;

(1014) if($d' < C(q)$) {

(1015) if(q is in $L1$) $\text{remove}(L1, q)$;

(1016) $C(q) = d'$;

(1017) $\text{ptr}(q) = p$;

(1018) $\text{push}(L1, q, d')$;

(1019) }

(1020) } //end of for

(1021) if(flag_threshold) {

(1022) $\text{push}(L2, p, C(p))$;

(1023) }

(1024) } //end of inner while

(1025) $T = T + \delta_T$;

(1026) Copy $L1$ from $L2$ and clean $L2$.

(1027) } //end of outer while

6/6

FIG. 12

1200

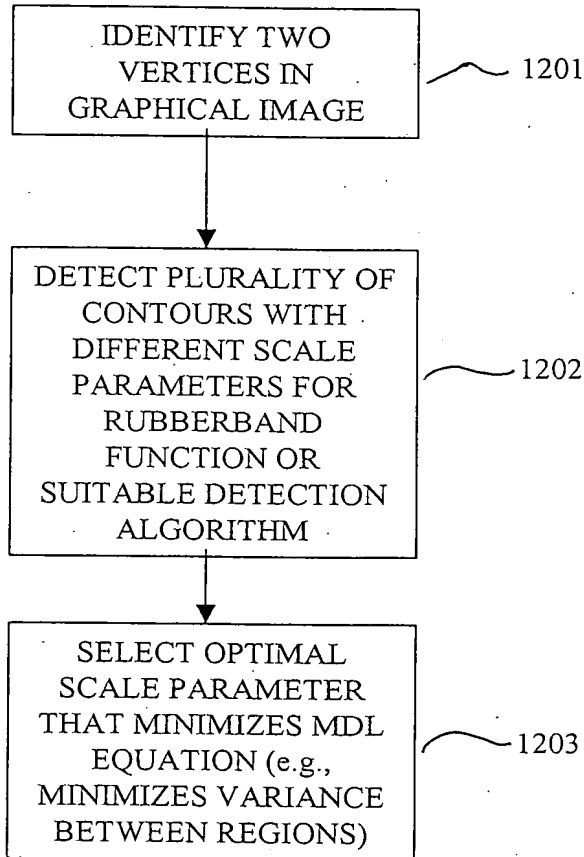


FIG. 12